## Chunking Files in Pandas – Part 1 (20 Points)

In this project, you will use Panda’s to process the data from the MinneMUDAC 2016 competition Dive into Water Data. The data can be found at the [MinneMUDAC site](http://minneanalytics.org/minnemudac/data/). You should document your work in a Jupyter notebook, which will be used to submit your solution.

1. Download all of the data in csv file format. Use !head -n 5 file\_name to inspect one of the files. Note that the files are separated with pipes, i.e. |.
2. First, you will explore the columns of each file. These files are pretty large, so you will need to use Pandas chunksize. Our first task is the exploration of the column labels. Pick two of the files and do the following:
   1. Read in the first chunk of each file. Remember that you need to use read\_csv with chunksize=500, sep='|', and toolz.first.
   2. Turn each of the columns into Python sets. Use the set union, intersection, and difference to answer the following questions: Are the columns the same? If not, which columns are in common?
3. Now you need to make a list of file names. The easiest way to do this is using a list comprehension and string formatting. Make a base file name string with a “hole” where the year goes. Then using range and the string format method to create a list of file names.
4. Now we are going to make a list of Pandas read\_csv iterators. Use a list comprehension to iterate through the list of file names and apply read\_csv with chunksize=500 and sep='|'. (If your program stalls, you probably forgot the chunksize!)
5. Now we will extract the first chuck from each file. Use toolz.first in a list comprehension to create a list of Pandas data frames, each consisting of the first chunk of the perspective file.
6. Now we are going to make a list of column name sets, one for each file. Use a list comprehension, df.columns, and the set constructor to create a list of column name sets.
7. Finally, we want a list of the common column names. You will need to use the accumulator pattern to accomplish this task. The initial value of the accumulator is the empty set, i.e. set([]), and you will update the accumulator by taking the union of the accumulator and the next column name set.
8. Finally, let’s determine which files have extra columns. Use a list comprehension on the zipped file names and column name sets. Use a filter that only keeps the entries with more columns (len will be helpful here). Keep a tuple with the file name and number of extra columns.